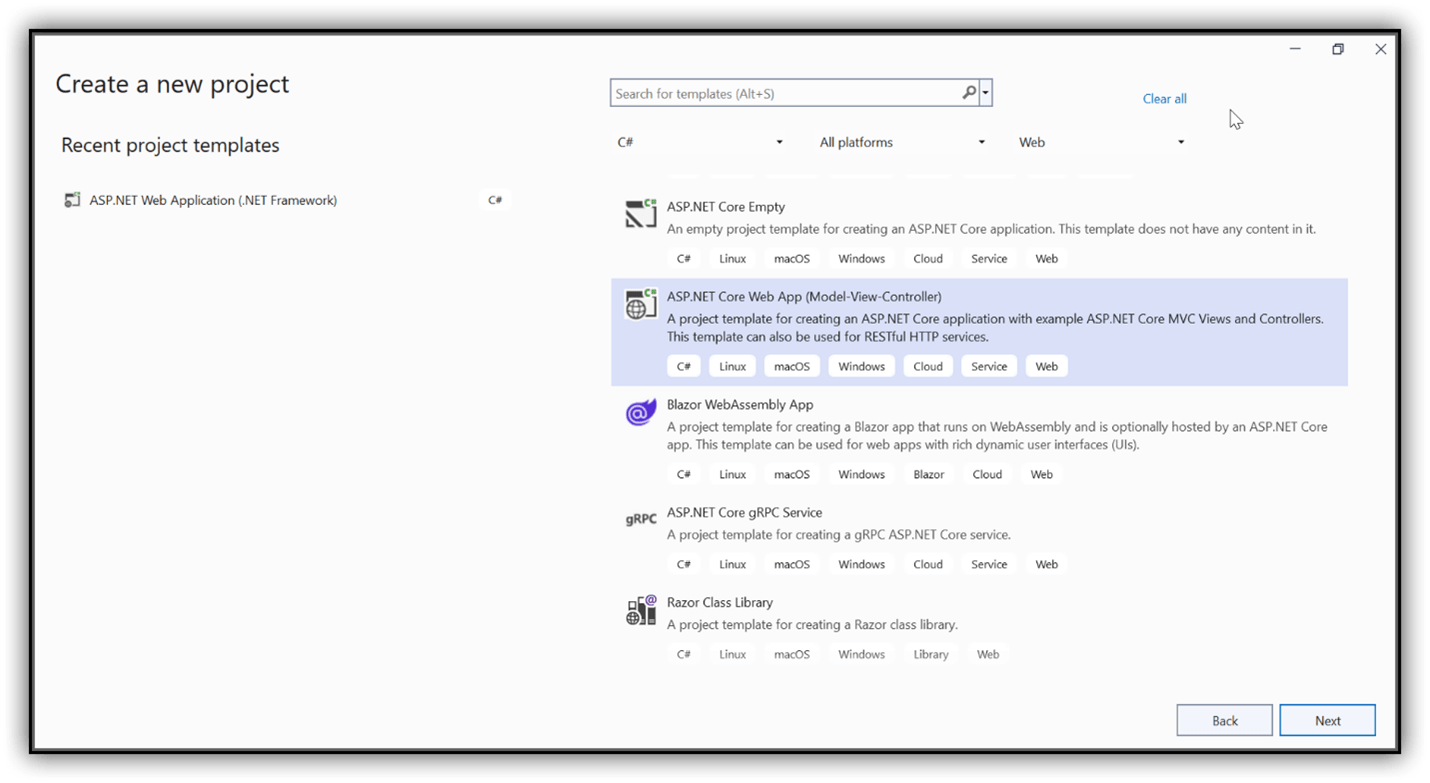
**How to convert PDF to Word in the ASP.NET Core platform**

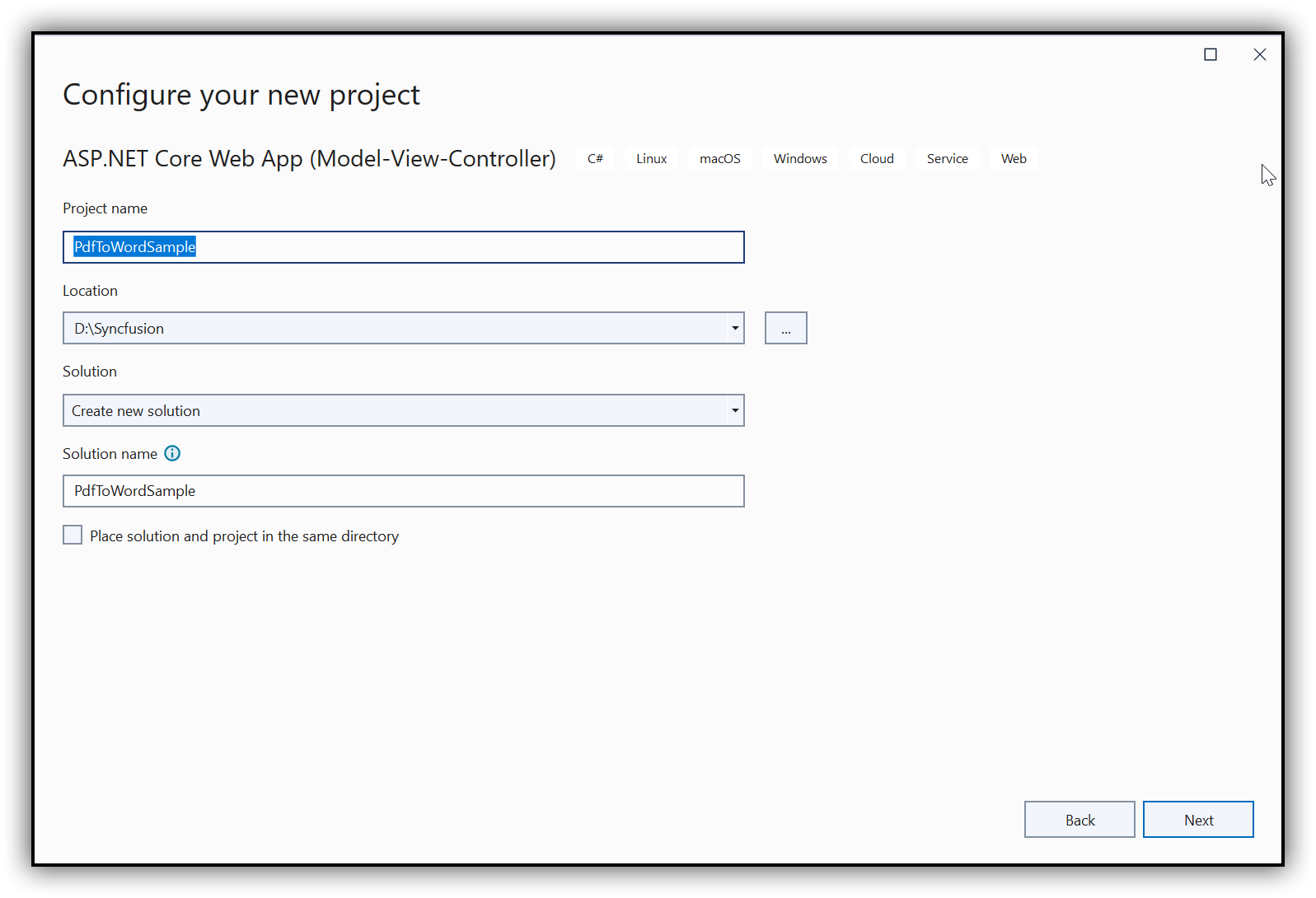
Essential® [ASP.NET Core PDF](https://www.syncfusion.com/document-processing/pdf-framework/net-core) does not have native support for converting PDF to Word documents. However, as a workaround, the PDF document can be converted to a word document by exporting the PDF pages into images, and then, the exported images can be added to the word document using the DocIO library.

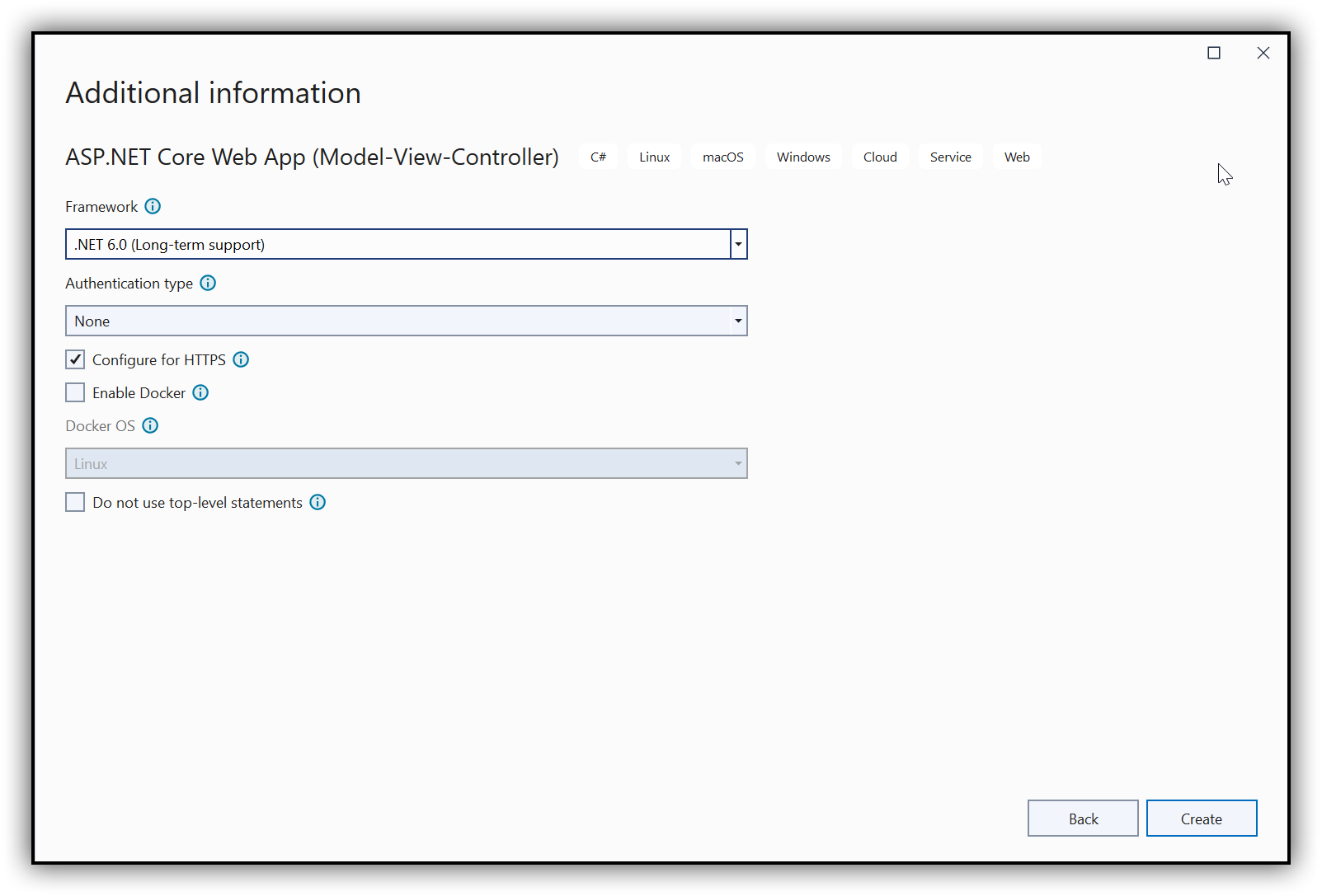
Steps to convert PDF to Word document programmatically:

1. Create a new ASP.NET Core web application project.



1. In configuration windows, name your project and select Next.





1. Install the [Syncfusion.EJ2.PdfViewer.AspNet.Core.Windows](https://www.nuget.org/packages/Syncfusion.EJ2.PdfViewer.AspNet.Core.Windows/) and [Syncfusion.DocIORenderer.Net.Core](https://www.nuget.org/packages/Syncfusion.DocIORenderer.Net.Core/) NuGet packages as a reference to your .NET Core application from [nuget.org](https://www.nuget.org/).

Syncfusion.EJ2.PdfViewer.AspNet.Core.Windows nuget image


Syncfusion.DocIORenderer.Net.Core Nuget image
 

1. Add a new button in the index.cshtml as follows.

**CSHTML**

<div class="btn">

    @{

        Html.BeginForm("ExportToPDF", "Home", FormMethod.Post);

        {

            <input type="submit" value="Export To PDF" class=" btn" />

        }

    }

</div>

1. A default controller with the name HomeController.cs is added on created the ASP.NET Core project. Include the following namespaces in HomeController.cs.

**C#**

**using** Microsoft.AspNetCore.Mvc;

**using** System.Drawing;

**using** Syncfusion.DocIO.DLS;

**using** Syncfusion.Pdf.Parsing;

**using** Syncfusion.EJ2.PdfViewer;

**using** SkiaSharp;

**VB.NET**

**Imports** System.Drawing

**Imports** Syncfusion.DocIO.DLS

**Imports** Syncfusion.Pdf.Parsing

**Imports** Syncfusion.EJ2.PdfViewer

**Imports** SkiaSharp

**Imports** Microsoft.AspNetCore.Mvc

1. Add the new method ExportToPDF in the HomeController.cs and include the following code sample to convert the PDF to Word in the ASP.NET Core platform.

**C#**

//Create a new Word document.

WordDocument m\_wordDocument = **new** WordDocument();

//Add a new section to the document.

IWSection section = m\_wordDocument.AddSection();

//Set the page margins to zero.

section.PageSetup.Margins.All = 0;

//Add a new paragraph to the section.

IWParagraph firstParagraph = section.AddParagraph();

SizeF defaultPageSize = **new** SizeF(m\_wordDocument.LastSection.PageSetup.PageSize.Width, m\_wordDocument.LastSection.PageSetup.PageSize.Height);

string path = "Barcode.pdf";

Stream docStream = **new** FileStream(path, FileMode.Open);

//Load the PDF document from the given file path.

using (PdfLoadedDocument m\_loadedDocument = **new** PdfLoadedDocument(docStream))

{

    //Use the Syncfusion.EJ2.PdfViewer assembly.

    PdfRenderer pdfExportImage = **new** PdfRenderer();

    //Load the PDF document.

    pdfExportImage.Load(docStream);

**for** (int i = 0; i < m\_loadedDocument.Pages.Count; i++)

    {

         //Export the PDF document pages into images.

         SKBitmap bitmap = pdfExportImage.ExportAsImage(i);

         SKImage image = SKImage.FromBitmap(bitmap);

         SKData encodedData = image.Encode(SKEncodedImageFormat.Jpeg, 100);

         MemoryStream imageStream = **new** MemoryStream();

         encodedData.SaveTo(imageStream);

         //Add an image to the paragraph.

         IWPicture picture = firstParagraph.AppendPicture(imageStream);

        //Set width and height for the image.

        picture.Width = defaultPageSize.Width;

        picture.Height = defaultPageSize.Height;

        imageStream.Dispose();

    }

//Save the PDF to the MemoryStream.

MemoryStream stream = **new** MemoryStream();

m\_wordDocument.Save(stream, Syncfusion.DocIO.FormatType.Docx);

//Set the position as '0'.

stream.Position = 0;

//Download the PDF document in the browser.

FileStreamResult fileStreamResult = **new** FileStreamResult(stream, "application/msword");

fileStreamResult.FileDownloadName = "Result.docx";

return fileStreamResult;

}

VB.NET

'Create a new Word document.

**Dim** m\_wordDocument **As** WordDocument = New WordDocument()

'Add a new section to the document.

**Dim** section **As** IWSection = m\_wordDocument.AddSection()

'Set the page margins to zero.

section.PageSetup.Margins.All = 0

'Add a new paragraph to the section.

**Dim** firstParagraph **As** IWParagraph = section.AddParagraph()

**Dim** defaultPageSize **As** SizeF = New SizeF(m\_wordDocument.LastSection.PageSetup.PageSize.Width, m\_wordDocument.LastSection.PageSetup.PageSize.Height)

**Dim** path **As** String = "Barcode.pdf"

**Dim** docStream **As** Stream = New FileStream(path, FileMode.Open)

'Load the PDF document from the given file path.

**Using** m\_loadedDocument **As** PdfLoadedDocument = New PdfLoadedDocument(docStream)

  'Use the Syncfusion.EJ2.PdfViewer assembly.

**Dim** pdfExportImage **As** PdfRenderer = New PdfRenderer()

    'Load the PDF document.

    pdfExportImage.Load(docStream)

**For** i **As** Integer = 0 **To** m\_loadedDocument.Pages.Count - 1

        'Export the PDF document pages into images.

**Dim** bitmap **As** SKBitmap = pdfExportImage.ExportAsImage(i)

**Dim** image **As** SKImage = SKImage.FromBitmap(bitmap)

**Dim** encodedData **As** SKData = image.Encode(SKEncodedImageFormat.Jpeg, 100)

**Dim** imageStream **As** MemoryStream = New MemoryStream()

       encodedData.SaveTo(imageStream)

        'Add an image to the paragraph.

**Dim** picture **As** IWPicture = firstParagraph.AppendPicture(imageStream)

        'Set the width and height for the image.

        picture.Width = defaultPageSize.Width

        picture.Height = defaultPageSize.Height

        imageStream.Dispose()

**Next**

**End** **Using**

'Save the PDF to the MemoryStream.

**Dim** stream **As** MemoryStream = New MemoryStream()

m\_wordDocument.Save(stream, Syncfusion.DocIO.FormatType.Docx)

'Set the position to '0'.

stream.Position = 0

'Download the PDF document in the browser.

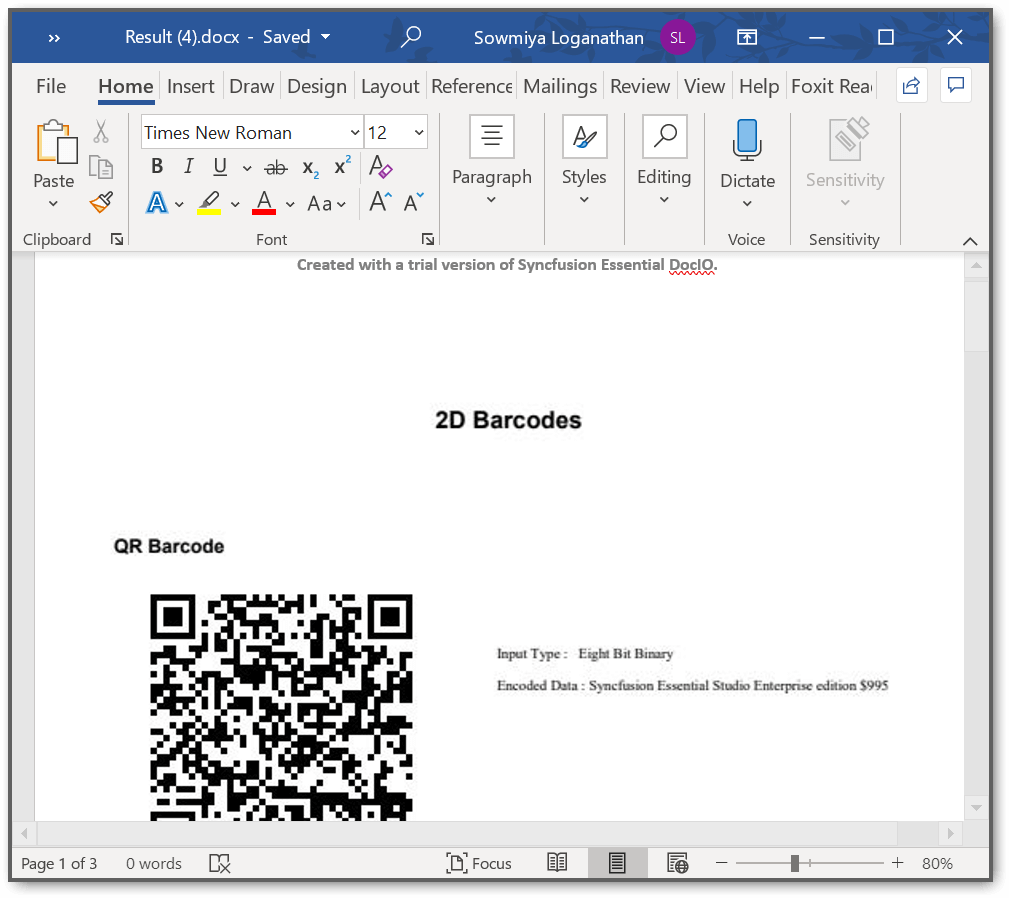
**Dim** fileStreamResult **As** FileStreamResult = New FileStreamResult(stream, "application/msword")

fileStreamResult.FileDownloadName = "Result.docx"

**Return** fileStreamResult

A complete working sample to convert PDF to Word in the ASP.NET Core platform can be downloaded from the [PDFToWordSample.zip](https://www.syncfusion.com/downloads/support/directtrac/general/ze/PdfToWordSample-83519269.zip).

By executing the sample, you will get the output document as follows.



Take a moment to peruse the [documentation](https://help.syncfusion.com/file-formats/docio/working-with-paragraph), where you can find the adding of text, images, lists, hyperlinks, and symbols in word paragraph. Explore more about the rich set of Syncfusion® [Word Framework](https://www.syncfusion.com/word-framework/net) features. An online example of [exporting PDF pages to image](https://ej2.syncfusion.com/aspnetcore/documentation/pdfviewer/how-to/export-as-image/)s.

Note:

Starting with v16.2.0.x, if you reference Syncfusion® assemblies from the trial setup or the NuGet feed, include a license key in your projects. Click this [link](https://help.syncfusion.com/common/essential-studio/licensing/license-key) to learn about generating and registering the Syncfusion® license key in your application to use the components without a trail message.

​**See Also:**[**Convert the PDF to Word in Windows Forms**](https://www.syncfusion.com/kb/8084/how-to-convert-pdf-document-to-word-document)

**Conclusion**

I hope you enjoyed learning about how How to convert PDF to Word in the ASP.NET Core platform.

You can refer to our [**.NET PDF Core’s feature tour**](https://www.syncfusion.com/document-processing/pdf-framework/net-core) page to know about its other groundbreaking feature representations and [**documentation**](https://help.syncfusion.com/file-formats/pdf/create-pdf-file-in-asp-net-core), and how to quickly get started for configuration specifications. You can also explore our [**PDF example**](https://www.syncfusion.com/demos/fileformats/pdf-library) to understand how to create and manipulate data in the .NET PDF.

For current customers, you can check out our [**Document processing libraries**](https://www.syncfusion.com/document-processing-libraries) from the [**License and Downloads**](https://www.syncfusion.com/account/downloads) page. If you are new to Syncfusion®, you can try our 30-day [**free trial**](https://www.syncfusion.com/downloads) to check out our JavaScript Context Menu and other JavaScript controls.

If you have any queries or require clarifications, please let us know in the comments section below. You can also contact us through our [**support forums**](https://www.syncfusion.com/forums), [**Direct-Trac**](https://www.syncfusion.com/support/directtrac/incidents/), or [**feedback portal**](https://www.syncfusion.com/feedback/aspnet-core). We are always happy to assist you!